



Original citation:

Joy, Mike and Luck, M.. (1998) Computer-based submission and assessment in BOSS. Interactions, Volume 6 (Number 2).

Permanent WRAP url:

<http://wrap.warwick.ac.uk/61045>

Copyright and reuse:

The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions. Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

A note on versions:

The version presented in WRAP is the published version or, version of record, and may be cited as it appears here. For more information, please contact the WRAP Team at: publications@warwick.ac.uk



<http://wrap.warwick.ac.uk/>

Computer-Based Submission and Assessment in BOSS



Mike Luck and Mike Joy, Computer Science, University of Warwick

The effects of ever-increasing student numbers continue to impact on the quality of course delivery in both further and higher education. This is especially true for computing courses in which there is a high practical programming component, and which are assessed by programming assignments. With large numbers of students, many lecturers are faced with the stark choice of investing vast amounts of time and effort in marking, through the separate processes of retrieving student programs, compiling them, running them and testing them manually against sets of test data, or of simply visually scanning a program and assigning a mark to it on the basis of a cursory assessment. The first of these options is becoming impossible due to the increasing demands on lecturers, yet the second is hardly acceptable.

By developing techniques for automating the submission, compilation and testing of student programs, we can support the process of marking, and enable marking tasks to be divided among several individuals while maintaining rigour and consistency. Not only can this stem the tide of an increasing workload, it can improve the learning experience available to students (in providing them with facilities for immediate and effective feedback, for example,) and also enable other administrative tasks to be automated as part of a coherent approach to full course management.

The BOSS system for automatic submission of assignments, built in an effort to address the problems described above, comprises a collection of programs, each of which performs a different task contributing to the overarching goal of effectively managing the process of submitting programming assignments on-line. BOSS is designed specifically for courses with large numbers of students, assessed by means of programming exercises. The individual components of BOSS are designed not be used by two kinds of individual. First, some programs must be used by students so that they can gain feedback and submit their programs. Second, lecturers and any course tutors involved in assisting the lecturer must be able to gain access to the submitted programs in order to test and mark the student submissions. The programs offer the following functionality.

Students may submit programs on-line by means of a user-friendly program that conducts a dialogue with the student to ensure that the correct submission is made. The program is stored and simple checks are carried out (to ensure the correct programming language is used and to verify the student's identity, for example), so that the lecturer can subsequently test and mark it.

All submissions for a specified item of coursework can be run against a number of sets of data. The output from the students' programs are compared with the expected output for each set of data. Time and space limits are placed on the execution of a program so as to prevent a looping program from continuing unchecked, and other steps are taken to minimise the potential for a program to damage the system.

Submissions and the results of the testing process can be inspected on-line by authorised staff. Anonymity is preserved by storing data by University ID number.

Students can test their programs by running them against one data set on which they will eventually be tested, and under precisely the same conditions. This ensures that the program will work as the student expects when being tested and marked. In addition, it provides students with confidence that their submitted work does pass some minimal requirement.

Final marks are stored in a SQL database and correlated with information from the University database (names and courses versus ID numbers and course registration, for example) to produce final marksheets for examination secretaries.

The BOSS system is a tool to allow students to submit assignments, and for those programs to be tested automatically. It is not an automated marking system. It is the responsibility of the individual lecturer to provide a marking scheme which takes account of the results produced by BOSS, together with all other factors which may be regarded as important (such as program style, commenting, etc.).

We do not aim to remove the instructor from the teaching loop, but instead simply to assist the instructor in achieving a quicker, more accurate and more consistent assessment of programming assignments. This is important, and should be made clear to students to avoid any misconceptions about the extent and scope of the

automated system. It is our experience that students gain confidence from the system, but they are also uneasy about the possibility of its unlimited significance in the assessment process.

The system we have running has been used for five years, albeit in different versions. It has been deployed on several courses, including those covering Pascal programming, UNIX Shell programming and C++ programming, each course attracting up to 200 students. As it stands, the system is functioning well. There has been a generally favourable student response, and this has improved as the culture of automatic submission has become established within the Department. In addition, lecturers and tutors have also found the system to be simple and easy to use, and marking times have been reduced significantly with a corresponding increase in consistency throughout.

During the five years the system has been used on our courses, we have enhanced the software in several ways. The original command-line interface to the system has been replaced with a graphical user-interface for both students and tutors, further extending the user-friendliness and functionality of the system as a whole. Not only does this provide students with a more intuitive means of submitting their assignments on-line, it also enables more aspects of the administration of assignments to be integrated into the system. The inclusion of electronic marksheets, for example, enables the provision of a system that supports both anonymous marking and double marking in a coherent, secure and efficient way.

In all these developments we have been concerned to ensure that the tools we use on our courses do not cause problems in the learning process. Moreover, no developments are intended to have a deleterious effect on the the learning opportunities that are available and an important part of our teaching strategy. Thus all intended future developments are simply designed to enhance the effectiveness of staff in providing a quick and effective service for students. Such possible future developments include an extension to the marking system to provide further information to the tutor to assist in other aspects of marking concerned with programming style, as well as making use of the on-line student mark information to provide detailed statistical analysis of courses and assignments for more effective course and student monitoring. In extending the system in this way, we aim to progress along the path towards a full course management facility.

Dr Mike Luck & Dr Mike Joy

Department of Computer Science

University of Warwick

Email: m.luck@warwick.ac.uk or m.joy@warwick.ac.uk
